

1N-77-2L
107057

P. 43

Automation of the Targeting and Reflective Alignment Concept

A Final Report
for Grant
NAG 9-374
9/89 - 8/92
(RF 6344)

Submitted to:
Leo Monford

NASA Johnson Space Center
Mail Code IC
Houston, TX 77058

By:
Robin C. Redfield

Mechanical Engineering Department
Texas A&M University
College Station, TX 77843-3123

October, 1992

N93-12959

Uncles

N2/90 0129697

(0792-017904) AIRCRAFT DAMAGE FROM
SMALL ASTEROIDS (LARGE) 13 P

Table Of Contents

ABSTRACT	1
INTRODUCTION	1
BACKGROUND FOR THE AUTOMATION	3
Coordinate System	3
Alignment	4
Equipment	6
AN OVERVIEW OF THE AUTOMATION STRATEGY	7
DEVELOPMENT OF THE SENSOR	8
DEVELOPMENT OF SOFTWARE	13
Distinguishing Target from Surroundings	15
Interpreting Mirror Position with Respect to Camera	17
Control of Manipulator	23
Algorithms	24
The Complete System	31
PERFORMANCE OF THE AUTOMATION	32
Uncertainty Analysis	32
Range Uncertainty	34
Lateral Uncertainty	35
Rotational Uncertainty	35
Roll Uncertainty	36
Implementation of the Automation	36
CONSIDERATIONS AND LIMITATIONS	39
BIBLIOGRAPHY	41

Abstract

The automated alignment system, described herein, employs a reflective, passive (requiring no power) target and includes a PC-based imaging system and one camera mounted on a six degree of freedom robot manipulator. The system detects and corrects for manipulator misalignment in three translational and three rotational directions by employing the Targeting and Reflective Alignment Concept (TRAC) [11], which simplifies alignment by decoupling translational and rotational alignment control. The concept uses information on the camera and the target's relative position based on video feedback from the camera. These relative positions are converted into alignment errors and minimized by motions of the robot. The system is robust to exogenous lighting by virtue of a subtraction algorithm which enables the camera to only see the target. These capabilities are realized with relatively minimal complexity and expense.

Introduction

Alignment is a common engineering problem. In general, it requires that two objects be placed at a specific pose with respect to one another. Pose is a description of alignment that can include all six degrees of freedom (translational x , y , z ; and orientation roll, pitch and yaw). The roots of alignment are in the land surveyor's geodetic transit of the 19th century [13]. Techniques from 100 years ago may seem crude when compared with the high speed signal processing and exotic sensors available now, but the concepts, however, remain quite similar. In the past, the sensor was the human eye and the image processor was the human brain. By the early and middle 20th century, human optics were used for off-line manufacturing inspection and gaging. The tools of the trade included optical flats, microscopes and optical comparators [17]. More recently, the pressures of higher performance manufacturing gave birth to automated inspection and alignment.

This automation of alignment has many applications including manufacturing, docking of space vehicles, underwater vehicle manipulations, hazardous environment manipulations and shaft alignment. Today's methods of alignment employ triangulation sensors, computer vision systems, laser interferometric techniques, and ultrasonic techniques. These methods are expensive and require high-tech hardware but save tremendous amounts of time and are much more accurate than their predecessors.

Many current alignment sensing systems use multiple sensors and triangulation techniques. Triangulation techniques use two or more sensors and knowledge of the sensors' relative positions. Saint-Marc, Jezouin and Medioni [15] present a triangulation-based range finding system that employs two cameras that look at a scene illuminated by a sheet of laser light. The automated alignment method described by Shi, Monchaud, Prat, and Domey[16] includes two cameras and an infrared emitter on the object to be located. The two cameras locate the position of the target in their two dimensional image plane. Knowledge of the cameras' positions relative to each other and the emitter's position in each image allows for determination of the position of the emitter in three dimensional space. The cameras can then be manipulated to keep the emitter in the center of their individual images.

Laser interferometric techniques provide very accurate alignment but are expensive and provide less than six-axis alignment capability. The basic principles of laser interferometry are described by Farnum[5] along with a discussion of its strengths and weaknesses. Ishikawa[8] and Laguesse[9] describe two of many fiber optic-based range finding sensors. Several shaft alignment concepts measure two translational and two rotational axes ([1], [2], [3], [12], [13]). Hamar and Gettelman[6], and Callari [4] describe the use of laser optics for machinery alignment with the laser as a straight-edge.

The hole sensor described by Stevenson[17] employs an emitting fiber that is focused on a target surface creating an illuminated spot. A detecting fiber receives the light reflected from the target. As the illuminated spot sweeps across the hole, the reflected light's intensity decreases at the first edge and increases again when the other edge is encountered. This method provides ample accuracy but is limited to two-dimensional translational alignment.

Ultrasonic sensors offer one-dimensional position measurement but provide reliability, wide measurement ranges, and insensitivity to dust, dirt and external lighting [10]. Robinson[14] describes an ultrasonic proximity sensor that measures range alignment error at a range of 4 to 256 inches and outputs an analog signal based on the error.

The National Aeronautics and Space Administration / Johnson Space Center (NASA/JSC) has approached Texas A&M University with the need to automate an alignment concept that they have developed to position the Space Shuttle's Remote Manipulator System's (RMS) end-effector at a

specific alignment with respect to an object to be grappled (e.g. a disabled satellite). This problem may be trivial on Earth where a supercomputer can be used to perform pattern matching routines or other complicated algorithms but this is not realistic on the Space Shuttle. It's on-board computer is in the process of being upgraded to a system comparable in speed to an 80286-based computer which will provide limited computer power.

Another consideration is fault tolerance. If the software system fails and the RMS locks in the unstowed position, the shuttle cannot return to Earth. The astronauts must be able to take manual control the RMS and accomplish the desired task or maneuver the RMS back into the payload bay. The previously discussed sensors provide very accurate alignment capabilities but they all require either too much computer power or are not easily made fault tolerant.

This thesis describes an automated tracking, targeting, and control strategy that uses the Targeting and Reflective Alignment Concept (TRAC), which is a highly sensitive but simple sensor for general alignment purposes. Some major components of the proposed strategy are currently being used by astronauts at the Manipulator Development Facility at NASA/JSC. This sensor detects and corrects for alignment errors in the three translational and three rotational directions for relatively little complexity. Because the sensor uses simple video feedback, if the system fails the astronauts can perform the algorithm without the computer software. Research has been done to develop TRAC as an automated alignment system, demonstrate its accuracy and precision, and examine the optical and geometric considerations that are necessary for it's implementation.

Background for the Automation

TRAC employs a camera on a manipulator and a reflective target (mirror) on a "target" object. The camera is used to provide video feedback of 1) the target and 2) the camera's reflection from the target. The concept can be demonstrated with a camera and a mirror.

Coordinate System

Before the alignment strategy can be discussed, a TRAC-based coordinate system must be described. The cartesian coordinate system of Figure 1 will be used, with a traditional definition of roll, pitch and yaw.

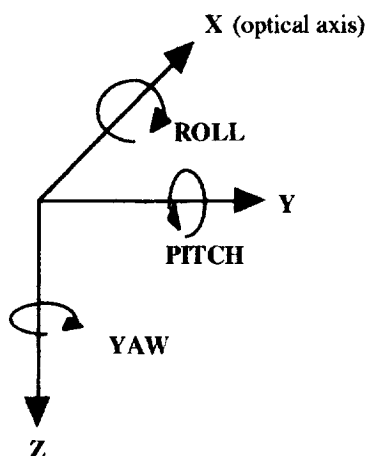


Figure 1 - Reference Coordinate System

Range is motion along the camera's optical axis, which is the x -axis in Figure 1. Motion in the y and z direction is referred to as lateral motion. Roll, pitch and yaw are defined as a rotation about the x , y and z axes, respectively.

Alignment

With the TRAC concept, alignment occurs when the camera's optical axis is normal to the reflective target and when the optical axis points to the center of the target. See Figure 2.

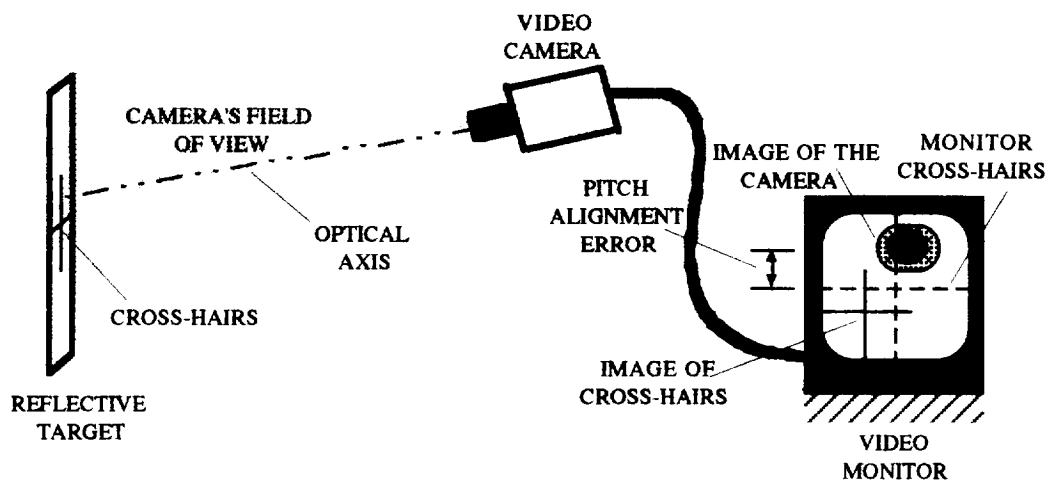


Figure 2 - Demonstration of Pitch/Yaw Misalignment

When the camera is normal to the target, the camera image in the video monitor is centered (The camera "sees" itself). When the optical axis points to the center of the target, markings on the reflective target, such as cross hairs, are also centered in the video monitor. This is described as rotational and lateral translational misalignment separately as follows.

Pitch and yaw misalignment occur when the camera's optical axis is not normal to the surface of the target. Remember that the camera is fixed to the manipulator. Misalignment is sensed when the camera's image appears off-center in the video monitor as demonstrated in Figure 2. For correct pitch and yaw alignment, the camera must be rotated about the target until its reflected image is centered in the monitor. This occurs when the optical axis is normal to the mirror. In Figure 2, the yaw error is minimal since the camera image in the monitor is nearly centered on the vertical monitor cross-hairs. To correct the pitch alignment error a positive pitch motion of the camera (manipulator) must occur.

Lateral misalignment is evident when the target is not centered on the camera's optical axis. It is sensed when the target fixture's image appears off-center (solid cross hairs) in the video monitor as demonstrated in Figure 3. The camera is normal to the target but the optical axis is not aligned with the cross hair center.

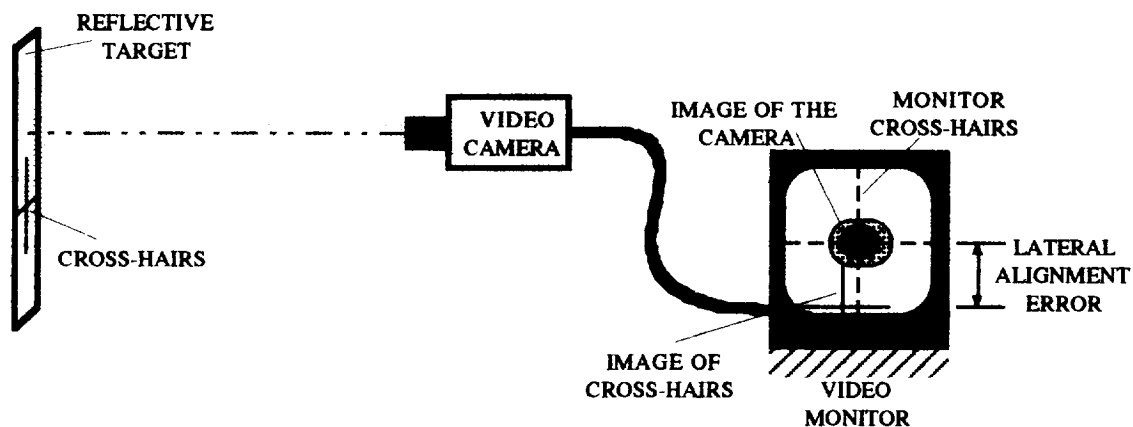


Figure 3 - Demonstration of Lateral Misalignment

In Figures 2 and 3 the centroid of the target is represented by the cross-hairs image on the monitor. The distance from the cross-hairs to the monitor's center represents the translational alignment error.

Note that translational alignment does not affect rotational alignment but the opposite is not true. Rotational alignment does affect translational alignment unless the axis of rotation is about the target cross hairs center. If the camera rotates about the cross hairs, the optical axis continues to align with the cross hairs center. If the camera rotates about any other axis, the rotation will cause the optical axis to move away from the cross hairs center. Because of this, it is easiest to first align rotation such that the camera is normal to the target and then align translation. Translation will not affect the previous rotational alignment. It is still beneficial to rotate as nearly as possible about the cross hairs center so that the camera does not "lose" the mirror upon rotation.

Equipment

An Hitachi Model Kp-120U Closed Circuit Television camera with a 19 mm lens is mounted on the end effector of a Unimate Model 560 PUMA manipulator for the automation of TRAC. The robot is controlled by a 80386-based computer. Low-level C callable routines have been developed that interface with the PUMA's teach pendant to perform the necessary motions. "Teach pendant" is another name for the manual control device for the robot. It allows a user to control the robot's motion with respect to different coordinate frames by activating button relays. In parallel communication with the PC is an Imaging Technologies Series 151 imaging system with MicroSoft C-based software from Mnemonics Inc.

To better understand this report, a description of some basic operations of imaging system is in order. Image processors digitize analog signals from cameras where the image plane of the camera is broken into a tiny grid. Each square section of the grid is referred to as a pixel. Each pixel location is assigned an output voltage based on the intensity of the light that it senses. The imaging system converts each of these voltages to a gray value between 0 and 255, with 0 representing the most dark and 255 representing the most light.

A binary threshold operation on an image transforms all pixels with a gray level of greater than a specified threshold white and all pixels with a gray level less than a specified value black. If the threshold is set to a high value then the brightest parts of the image will stand out. If the threshold is set low then the dark portions stand out.

In addition, the software from Mnemonics provides a "blob analysis". The blob analysis determines the centroid area and location of each "blob" in an image. A blob is defined as the set of

adjacent pixels that are within a specified range of gray values.

An Overview of the Automation Strategy

A concept for the alignment strategy must be formulated to develop the automation. This design takes into account several observations.

1. Positioning the camera so that the target is in the camera's field of view is a relatively simple task for the user to perform. The manipulator is controlled such that the camera's optical axis points towards the target.
2. Positioning the camera in such a way that it's own reflection can be seen in the mirror is a more difficult task for the user but is a prerequisite to implementation of the automation. The optical axis must be nearly normal to the target mirror.
3. Because the system is being designed for use on the Space Shuttle, the user is ultimately in control of the automation, he is "in the loop". The automation will be designed to retain manual supervision and emergency override capabilities.

The first two observations have been shown to be true by astronauts at the Manipulator Development Facility (MDF) at JSC when attempting to manually implement TRAC. Conceptually, a logical outcome of these observations is the development of two alignment strategies to achieve the final precision alignment. These are referred to as the gross positioning and fine positioning algorithms. Their relation is shown in Figure 4.

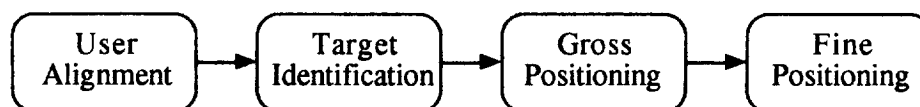


Figure 4 - General Flow of the Alignment Algorithm

To perform the automation, a user manually positions the end-effector of the manipulator so that the camera is within viewing distance and oriented towards the target mirror. The imaging system recognizes the target (as will be detailed shortly) and the user verifies the identification. The

relative rotational misalignment and the translational misalignment of the target in relation to the end effector is then determined. The gross positioning algorithm is used to approximately align the camera normal to the target mirror until the camera can see its own reflection. The fine positioning algorithm is then used to accurately align an object in order to grapple, contact, or perform insertion tasks. This two tiered approach is necessary because fine pose control requires the reflection of the camera to be seen in the camera's field of view. Relatively small misalignment between the camera's optical axis and the normal target mirror results in the image of the camera being out of its own view.

To implement the gross positioning algorithm, a method must be developed to obtain information about the camera's and target's relative positions when large errors occur in the pitch and yaw positions. This algorithm must perform this task with a high reliability. The fine positioning algorithm will use the methods of TRAC to determine and correct pitch, yaw and lateral alignment errors.

After the camera is normal to the target and centered on the cross hairs, two additional degrees of freedom remain, range and roll. Range control is guided by the apparent size of the target. For manual TRAC this is the perceived size of the cross hair grid. Roll control is guided by a given coordinate of on the target as "up." For manual control, one axis of the cross hairs is marked to identify it from the others.

Development of the Sensor

The automation requires that a sensor be developed that employs TRAC for fine positioning but also allows for the gross positioning to be accomplished. In short, a target must be developed that reveals information about its position with respect to the camera. The target must have special characteristics to enable the vision system to easily and quickly recognize it.

A traditional means of target recognition is template matching in which the target has markings on it that the imaging system is programmed to recognize. Every pixel seen by the vision system must be checked against a template image when using this method. Obviously, this takes large amounts of computer time. Because in-flight computers on the Space Shuttle are relatively slow, template matching does not provide the speed to allow for precision real time control of the

RMS. Consequently, a faster method is needed.

In the initial stages of this research, schemes were developed for target recognition that employ very reflective or very non-reflective objects to mark the target and a binary threshold of the image to make the markings more prominent. Thresholding was chosen for testing because it is computationally inexpensive when compared to other methods currently in use.

In order to recognize the target with dark markings, a flat black paint (3M's Nextel Velvet Coating 101-C10 BLACK) was used to form cross-hairs to mark the center of the mirror as shown in Figure 5.

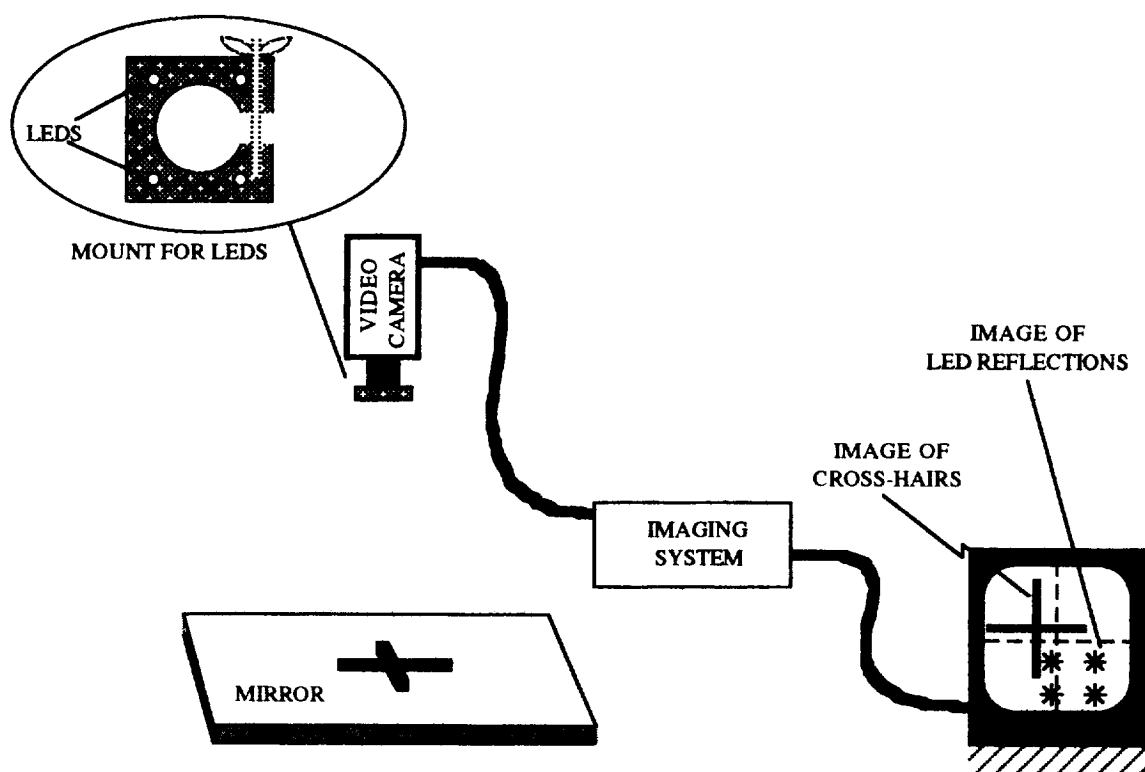


Figure 5 - Testing with Black Paint as Cross-Hairs

Four LEDs were placed in an aluminum mount around the edge of the camera's lens. Software was developed to perform two binary threshold operations on the image. One thresholding facilitates determination of the centroid of the cross-hairs and the other facilitates determination of the centroid of the LEDs. A binary threshold operation with a high value for the threshold is first performed on the image. This makes the bright objects stand out in the image. The centroid of these images is the centroid of the LEDs, which is a measure of the camera's position. The second threshold is about a

low value, which causes the dark portions stand out. The LEDs were easily recognized as long as no bright objects appeared in the camera's field of view. The paint, however, did not make the target easily identifiable. While the paint was always recognized as maximally dark by the imaging system, other objects were perceived to be just as dark. Therefore, target identification was compromised.

Along the same lines, it was conceived that a bright surface may be easily recognizable. 3M High Intensity Sheeting was next tested. This is a material used for the speed limit signs found on highways. It is a highly reflective material that provides its best reflectance when normal to the light source (97% reflectance within 3 degrees to the normal). The sheeting was tested as a cross-hair in a similar fashion to the black, painted cross-hair. The sheeting did not provide for as much contrast as expected.

From this experimentation another idea was spawned: Use the High Intensity Sheeting as the target and the black paint as the cross-hairs. When a light is shown on the target, the High Intensity Sheeting reflects most of the light but the paint absorbs most of the light that is shown on it. By thresholding out the bright part, the target's translational position is revealed by the cross-hairs. By mounting a light on the camera and thresholding out the dark part of the image, the camera's position in the image can be determined. The method proved to be good for recognition and tracking of the target but poor for camera recognition. The light mounted on the camera is only distinguishable if the camera's optical axis is very near perpendicular with the surface of the target.

While these procedures did not provide the ideal target, they did provide some good insight. The only reason this last method failed was because the camera could not be identified. A previous method failed because the target identification was poor.

Another device that is similar to the High Intensity Sheeting is a retroreflector. In fact, the High Intensity Sheeting is essentially many poor quality retroreflectors placed next to one another on a sheet. Retroreflectors reflect light parallel to their incident angle and at a slight offset based on the dimensions of the retro-reflector as shown in Figure 6.

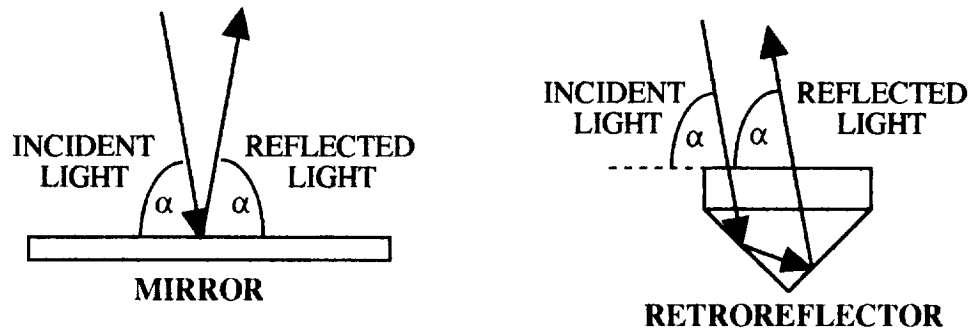


Figure 6 - Description of Retro-reflectors in Two Dimensions

When a light source shines on the target, the light that strikes the mirror reflects away at an angle equal to the incident angle. The light that strikes a retroreflector is reflected directly back to the light source. This was tested with the camera and mount as shown in Figure 7.

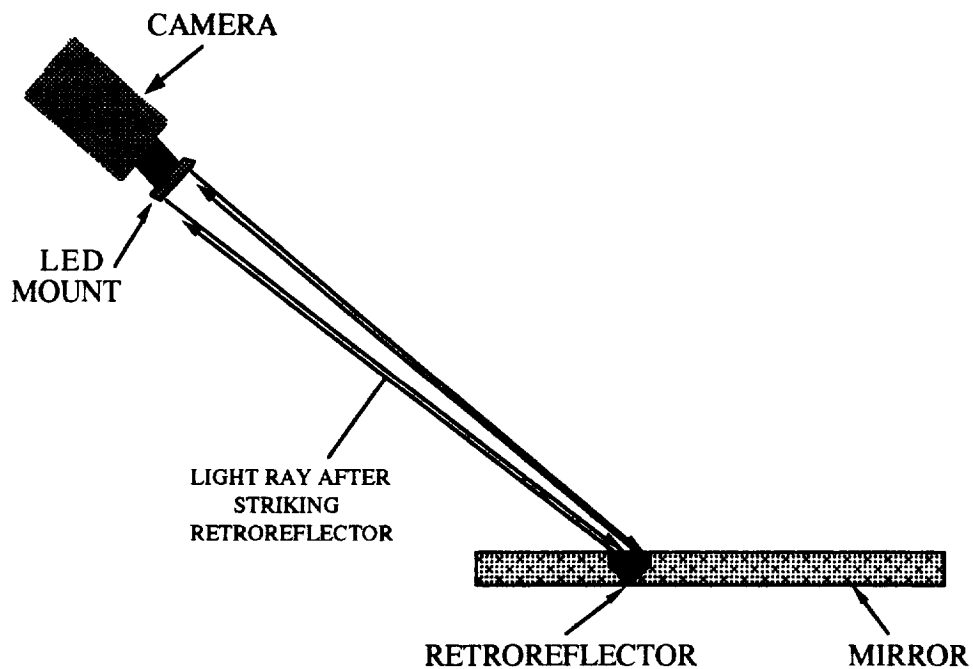


Figure 7 - Initial Camera and Target Set-up

A problem developed because of the precise directionality of the retroreflectors. The light from the retroreflectors actually misses the image plane of the camera. It is reflected back to the LED, not into or through the lens of the camera. For this reason the LED was wired over the center

of the lens at a slight offset as shown in Figure 8.

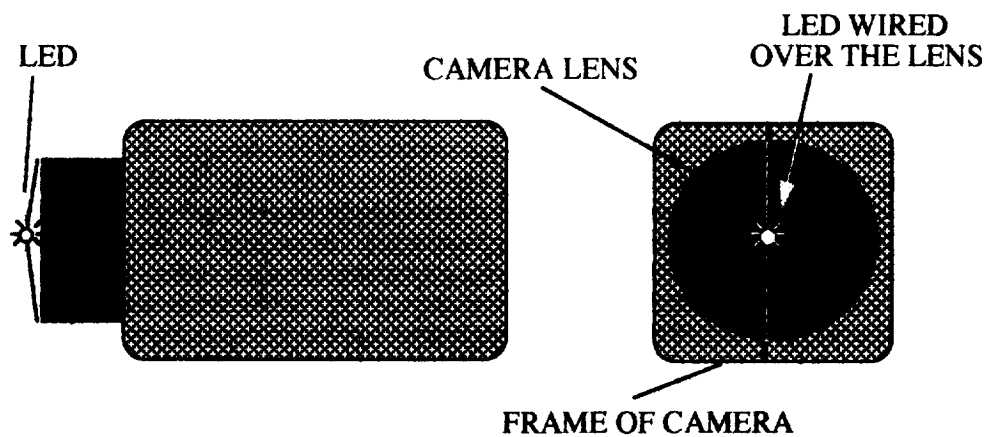


Figure 8 - Diagram of Camera and LED

When the light source is very near the center of the lens of the camera, this light appears in the retroreflectors in the camera's image. In Figure 9, the light source cannot be seen in the mirror but can be seen in the retroreflector. The camera is at too oblique of an angle to see its own reflection or that of the LED in the mirror.

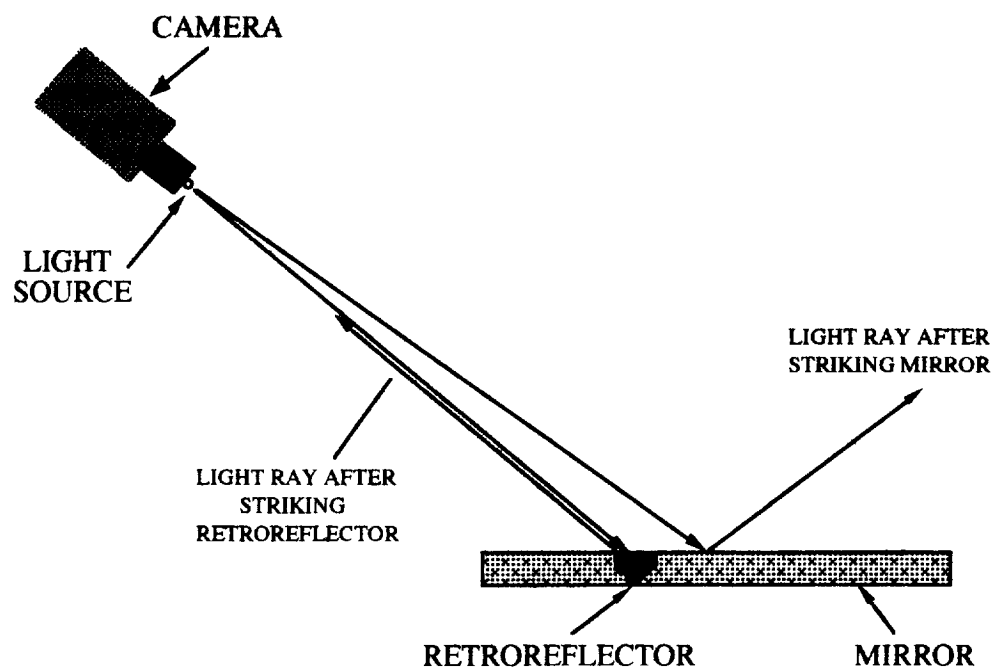


Figure 9 - Camera and Target Configuration

The light that strikes the mirror will be reflected away at the incident angle. The light that strikes the retroreflector will be reflected directly back into the camera's lens.

It should also be noted that the LED can be used to mark the camera in the image. The same LED used to illuminate the retroreflectors is used to locate the center of the camera's lens. The LED image's offset from the monitor center represents the pitch/yaw alignment error.

Three retroreflectors are necessary for determination of lateral displacement, range and orientation of the target, as shown in Figure 10.

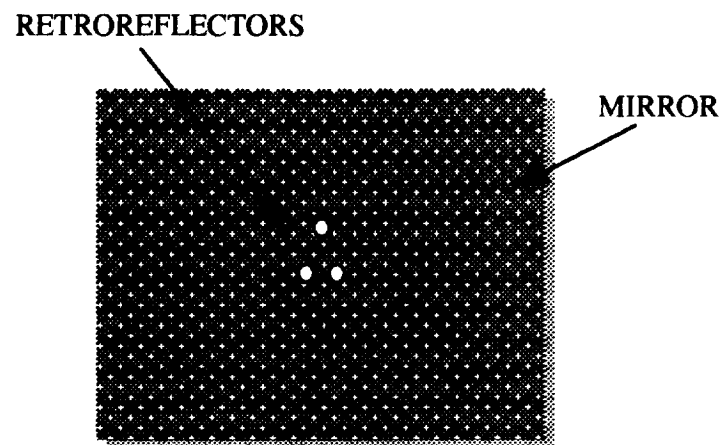


Figure 10 - Target with Retro-reflectors in the "Up" Position

If one retroreflector is placed at the center of the target then the lateral position of the target can be determined by measuring its offset from the center of the monitor, however there will be no indication of roll or range. If two retroreflectors are used then the range component of the target's position can also be computed. A calibration between the perceived distance between the retroreflectors and the range is used for this and is discussed in greater detail in the next section. Three retroreflectors in an isosceles triangle allow for roll alignment. To accomplish the roll alignment the target is oriented so that the triangle "stands up".

Development of Software

The automation is dependent on communication between the camera, imaging system, and manipulator. A communication diagram for the system is shown in Figure 11.

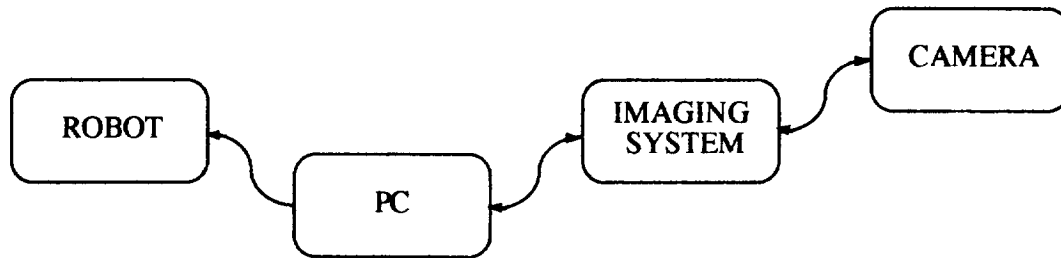


Figure 11 - Communication Diagram for the Automation of TRAC

The computer is the master in this system. It commands the imaging system to perform duties which include grabbing images, processing images, etc. The PC then interprets the information from the imaging system. This information is the basis for the commands given to the robot.

The communication allows for the determination of alignment errors and the subsequent nulling of these errors. The computer is connected in parallel with the imaging system to allow for high speed data transfer. The camera's image acquisitions are controlled by the imaging system and are stored in and processed by the imaging system. The PC then interprets the imaging systems data to determine the six positional errors. Once the errors are identified, they are nulled by implementing routines that command the manipulator to move in the positive or negative x , y , z , roll, pitch or yaw directions.

The computer's control of the PUMA is accomplished by connecting the PC's serial output through TTL level circuitry into the teach pendant of the robot. The TTL level circuitry provides a means of interpreting the computer's output and sending the correct voltage levels to the teach pendant. Once the data is interpreted, the connection to the teach pendant is used to close a relay, which is what occurs when a button is pressed by a user. This signals the robot to perform the desired action.

The communication simply provides the means by which the automation can take place. To utilize it, software must be developed to:

1. Use the vision system hardware to distinguish the target from its surroundings
2. Interpret the mirror's position with respect to the camera

3. Control the manipulator based on the alignment information

Distinguishing Target from Surroundings

Upon the completion of hardware development for the sensor, a problem is evident upon implementation. When the image analysis is performed, noise arises in the system in the form of exogenous lighting. When a light or bright object enters the camera's field of view, the imaging system has no way of distinguishing this from the LED on the camera or the illuminated retro-reflectors. To eliminate background light in this system a series of subtractions is used, as demonstrated in Figure 12.

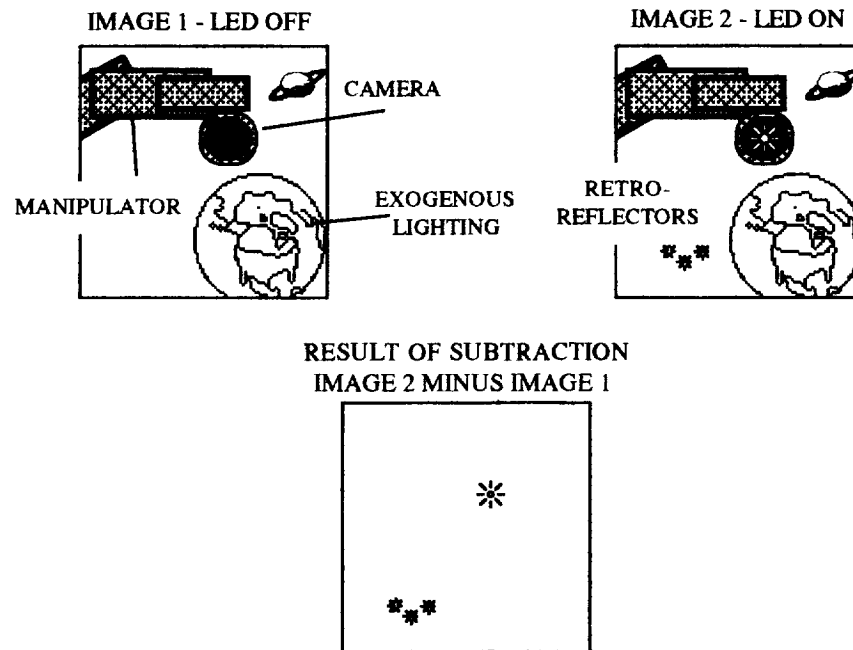


Figure 12 - Image Subtraction

The images shown are what the camera might see when zoomed in on a portion of the mirror. The first image is taken with the LED off. This image shows the camera, manipulator and any other objects behind the camera. The second image is taken with the LED on (the bright spot in the center of the camera). This image will include everything in the first image plus the reflection of the LED and the illuminated retro-reflectors. The "off" image is then subtracted from the "on" image. Subtraction of images is the literal subtraction of the gray scale values of corresponding pixels of

two images. The resulting image will include only the objects that have changed or are new since the "off" image. This will leave only the LED and the illuminated retro-reflectors in the subtracted image.

After the noise is reduced, the task of distinguishing the LED from the retroreflectors is simplified. The apparent size of the blob that results from the light reflected to the camera from the retro-reflectors as determined by the imaging system is a function of the retroreflector size and the position of the LED on the camera. If the retroreflectors perfectly reflected all of the LED's light parallel back to their source then the retroreflectors and the LED would appear to be the same size. This would be the case if the LED were placed precisely over the center of the lens. This cannot be done because the LED positioned over the center of the lens distorts the camera's view. Hence, the LED must be placed at a slight offset from the center of the lens. As the LED is positioned farther from the center of lens, the reflected light to the camera decreases as shown in Figure 13.

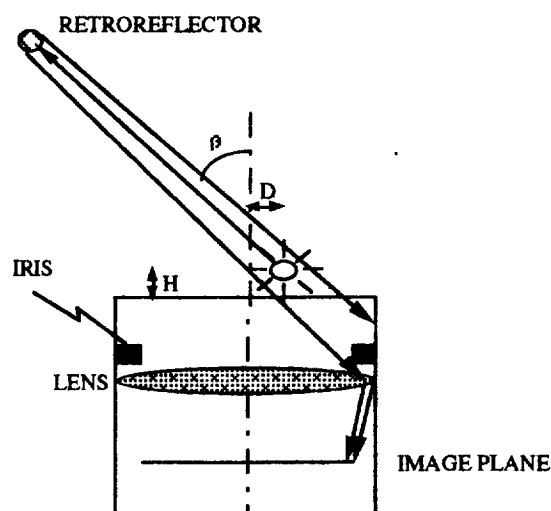


Figure 13 - Example of Iris Blocking the Light from the Retro-reflector

As the offset distances H and D are increased, the angle β also increases. This causes part of the reflection from the retroreflector to miss the lens. In the worst case, it may miss the lens altogether and be out of the camera's field of view even though the entire mirror is in the camera's field of view. Another factor is that the LED actually blocks some of the returned reflection onto the lens. As a consequence of these factors, the LED's reflection is always larger than the illuminated retro-reflectors. Thus the centroid of the largest blob that the vision system recognizes is the LED

which represents the centroid of the lens. The next three largest blobs are the retro-reflectors. Any additional blobs are considered noise in the system and are ignored.

Interpreting Mirror Position with Respect to Camera

When the camera's reflection is not visible in its own field of view little can be determined about the relative positions of the camera and target. However, if the target can be identified some information can be retrieved. (By target identification, it is meant that the retroreflector pattern can be identified by the imaging system.) If this is so, when the camera is rotated (pitch or yaw motion) about the mirror then the perceived image of the retroreflectors changes. In fact, it changes in a predictable manner. The perceived dimensions of the retroreflector pattern will be at a maximum when the camera's optical axis is perpendicular to the target's surface. The perceived dimensions will be at a minimum as the angle between the optical axis and the target approaches 90 degrees. This simple fact is used to implement the gross positioning or FIND algorithm that will be discussed in detail later.

When the camera's image is visible in the monitor, then the techniques of TRAC can be utilized. This provides for the prediction of the necessary motions in the y, z, pitch and yaw directions to make the camera perpendicular to the target. The concept requires that the LED and the centroid of the retroreflectors be positioned in the center of the video image for perpendicular alignment. Any deviation from this is considered an error in the direction of deviation. As produced by the imaging system these errors are measured in pixels. A method must be developed to convert this information into real spatial dimensions to make the information more useful for a control algorithm.

The method for conversion of the lateral errors is based on a conversion between pixels and actual spatial dimensions. It can be visualized with the aid of Figure 14.

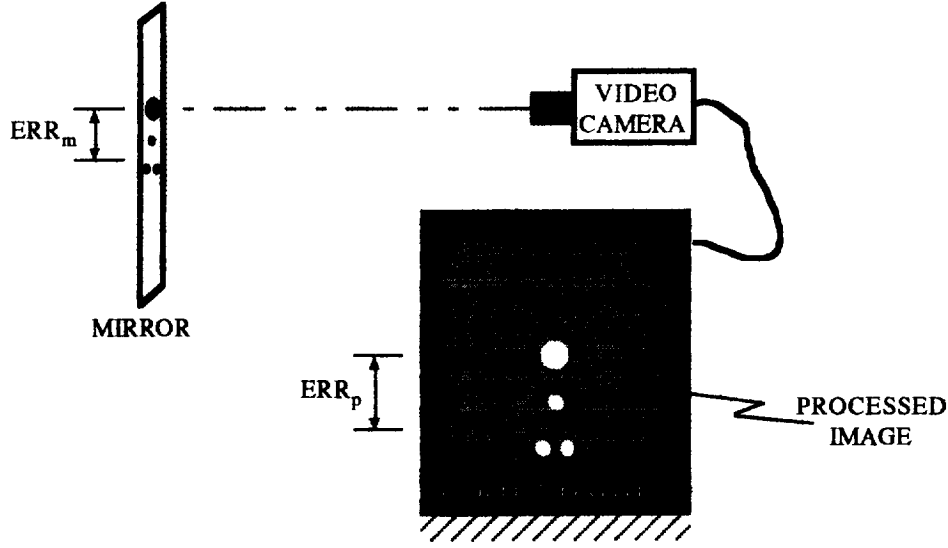


Figure 14 - Geometry of the Lateral Error Conversions

The method is really quite straight forward. ERR_m is the real spatial lateral alignment error measured in millimeters. The perceived lateral alignment error measured in pixels is referred to as ERR_p . ERR_m is the product of ERR_p and a conversion between the two. The most likely candidate for this conversion is the size of the pattern on the target. The perimeter of the isosceles triangle is of known dimensions and the perimeter of the triangle in the image is determined by the software. The ratio of these two values provides the required conversion factor.

$$ERR_m = ERR_p \left(\frac{PER_m}{PER_p} \right) \quad (1)$$

where PER_m and PER_p are the perimeter of the triangle in millimeters and pixels, respectively.

The key to the pitch and yaw conversion is the sensing technique developed for the range. The range is determined by use of a calibration for the size of the triangle formed by the retroreflectors. It is determined by an empirically derived curve fit for the relation between perceived perimeter size of the triangle pattern of retroreflectors and the real distance from the camera's lens to the target. The relation between the distance to the target and the perceived perimeter size is an offset hyperbola. This can be explained intuitively and by observations about the optics of lenses. At large distances, the size of the pattern does not appear to change very much with distance. At small distances, the change in pattern size is tremendous. In fact, before the camera lens touches the

mirror (range = 0), the entire pattern will no longer be in the camera's view; it will appear too large. This relation quickly brings to mind an inverse relationship between the variables and will be demonstrated with the help of the lens in Figure 15.

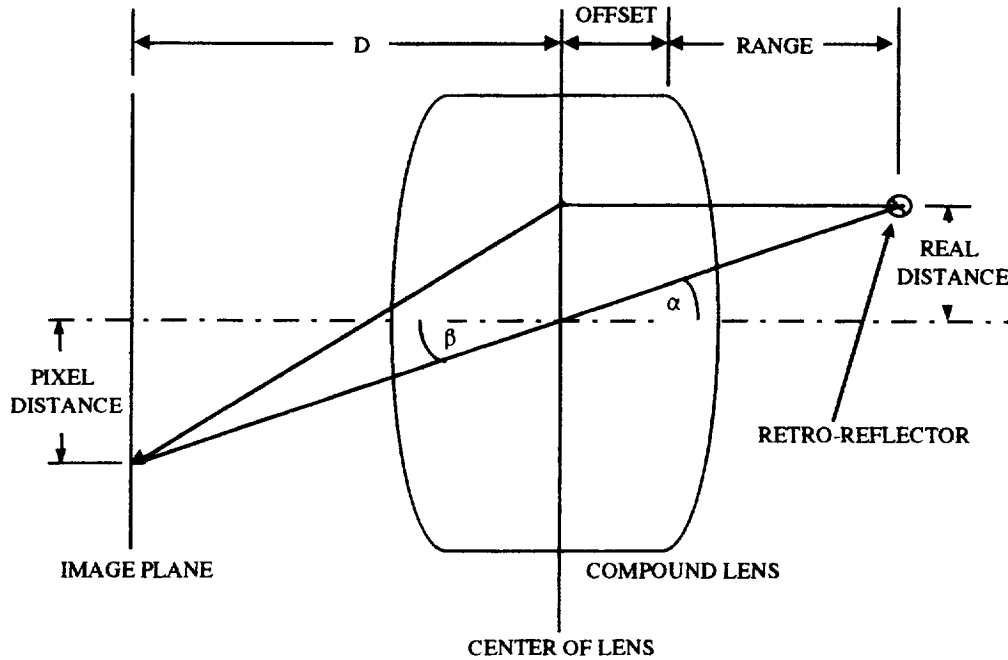


Figure 15 - Description of Range Equation

The angles α and β are equal. Therefore because of similar triangles the following equality is true.

$$\frac{\text{PIXEL DIST.}}{D} = \frac{\text{REAL DIST.}}{\text{OFFSET} + \text{RANGE}} \quad (2)$$

The offset is necessary because the range can only be measured from a known position- the front of the lens. The offset is the distance from the front of the lens to the center of the lens which is unknown. Solving equation 2 for the range produces the following.

$$\text{RANGE} = \frac{\text{REAL DISTANCE} \times D}{\text{PIXEL DISTANCE}} + \text{OFFSET} \quad (3)$$

where *REAL DISTANCE*, *D* and *PIXEL DISTANCE* are all constants. Therefore the following relation for the range can be used to produce a curve fit of the data where *A* and *B* are coefficients to be determined.

$$\text{RANGE} = \frac{A}{\text{PER}_p} + B \quad (4)$$

Software was developed to enable simple data acquisition at several distances to obtain data to determine A and B . These data are shown in Table 1.

Table 1 - Data for Determining Coefficients of Range Equation

PERIMETER (pixels)	RANGE (mm)
190.0	769.5
212.0	688.0
234.0	620.0
256.0	570.0
280.0	520.0
308.0	472.5
312.0	468.0
345.5	419.3
390.0	369.5
447.0	320.0
526.0	268.5
639.0	218.0

This was fit to an offset hyperbola and is shown with the actual data in Figure 16 with $A=149,349$. and $B = -14.55$.

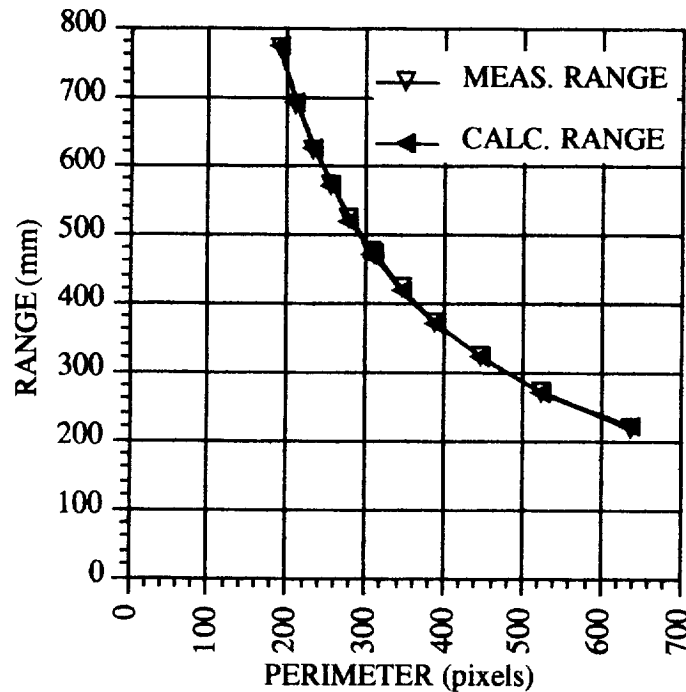


Figure 16 - Verification of Range Data Linear Regression

The range calculated with the values of A and B corresponds nicely with the range as measured. This range information is used to calculate the pitch and yaw positions of the target with respect to the camera. Figure 17 shows the geometry involved.

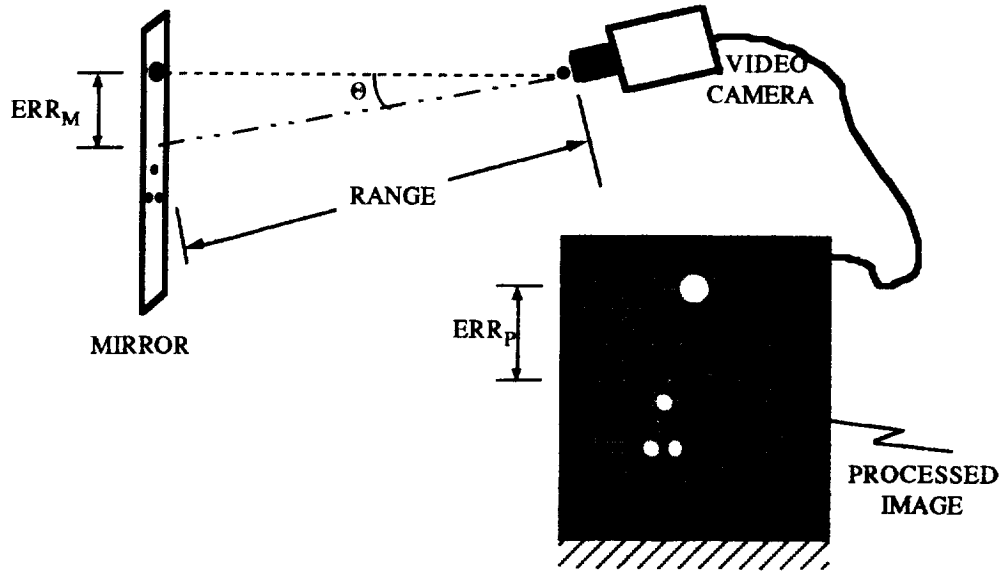


Figure 17 - Geometry for Real Pitch and Yaw Positions

In the figure, Θ is the real spatial rotational position of the camera (pitch or yaw) with respect to the target. ERR_m is the distance from the point where the imaginary perpendicular line between the mirror and the LED strikes the mirror (the optical axis) and the center of the mirror, measured in millimeters. ERR_p is the measure, in pixels, of the distance from the LED to the center of the monitor in the acquired image.

From the figure the following equality can be derived in much the same manner as the lateral alignment error is determined.

$$ERR_m = ERR_p \left(\frac{PER_m}{PER_p} \right) \quad (5)$$

The angle Θ is then determined by the equality,

$$\Theta = \sin^{-1}\left(\frac{ERR_m}{RANGE}\right) \quad (6)$$

The remaining alignment error to be determined is the roll error. The methodology to determine the roll position is only dependent on the retroreflectors position. Initially the second area moment of inertia of the image was used to calculate roll. For the triangle configuration this provides a unique value for each roll position but this proved to be computationally expensive.

A better scheme determines the roll error by first locating the centroid of the retroreflectors. The position of the *top* of the triangle is then determined. This is accomplished by determining the distance between each retro-reflector. The two that are closest together are the base of the triangle. The other one is the one to be aligned on top. Once the position of the centroid and the top retro-reflector are determined the roll position can be calculated as shown in Figure 18.

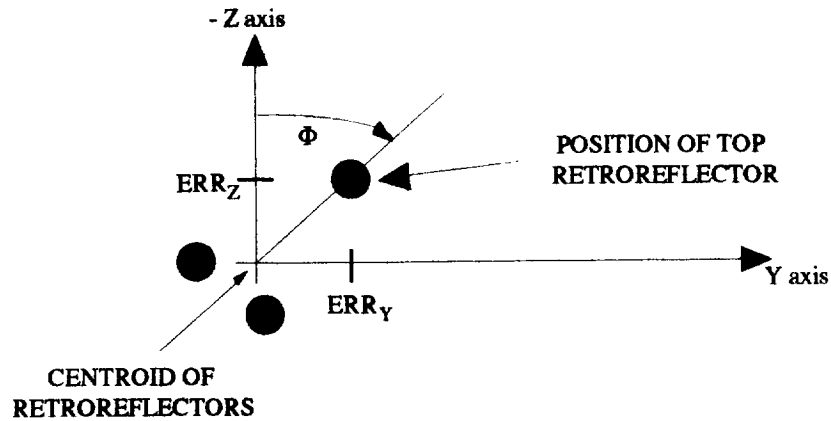


Figure 18 - Method for Determining Roll Error

The roll position is

$$\Phi = \tan^{-1}\left(\frac{ERR_Y}{ERR_Z}\right) \quad (7)$$

where

$ERR_Y = \text{TOP RETRO'S } Y \text{ POSITION} - \text{CENTROID'S } Y \text{ POSITION}$

$ERR_Z = \text{CENTROID'S } Z \text{ POSITION} - \text{TOP RETRO'S } Z \text{ POSITION}$

The relation for ERR_z may seem odd at first but remember that the angle Φ is from the negative z axis.

Control of the Manipulator

The next task is the development of a method to control the manipulator. A consideration in this development is a limitation of the interface with the PUMA; because the PUMA is designed to be "taught", only one axis can be commanded constant speed motion at any given time. In addition, when the subtraction algorithm is implemented, the camera must not be in motion. This is an inherent limitation of the subtraction algorithm and the speed of the image acquisition. The time between image acquisitions is approximately 33 milliseconds and this is enough time for the robot to move significantly between acquisitions. This results in two images that are not of identical scenes and therefore noise is introduced into the system.

Acknowledging these limitations, the simplest control is a pulse width modulated feedback scheme as outlined in Figure 19.

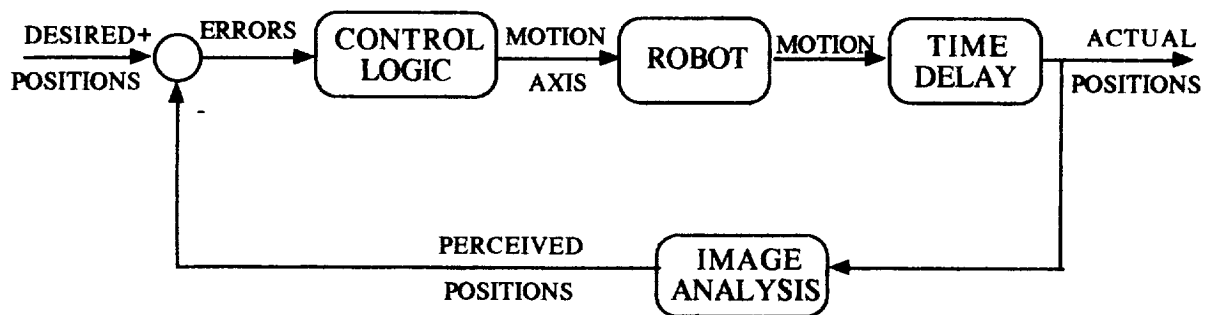


Figure 19 - Basic Closed Loop System

This basic outline shows how the system must logically progress. Six desired positions are compared with six perceived positions. These errors are passed to the control algorithm which decides which robot axes will be commanded to move. This information is then passed on to the manipulator which in turn causes the motion to occur. The controller then sends a signal for the motion to stop and the process starts again.

The time for each motion to occur is a simple matter to determine. It is the real spatial error in millimeter or degrees divided by the speed of the motion axis in millimeters per second or degrees per second.

$$\text{Delay} = \frac{\text{Error in Real Spatial Dimensions}}{\text{Speed of Robot Motion}} \quad (8)$$

For example, if a pitch error is sensed to be 5 degrees and the robot's pitch speed is 1 degree per second, then the robot will be commanded to perform a pitch motion for 5 seconds.

Algorithms

The algorithms used to fill the "control logic" block of Figure 19 must now be developed. As previously mentioned, the automation of TRAC is broken into two parts, gross positioning and fine positioning. The gross positioning algorithm is broken into three sub-sections. Initial target recognition, initial approach to target, and the "finding" of the camera's reflection. The initial target recognition is used to make sure the system can sense the mirror in its current pose. The initial approach to target is used to position the camera at a consistent starting position for the implementation of the FIND algorithm which drives the camera to approximate normal with the target. The fine positioning algorithm can be thought of as one procedure. These procedures are implemented as choices on a menu which will be described in greater detail later in this report. Before these procedures can be developed, the image processing procedures must be discussed.

The subtraction algorithm described previously is implemented as the flow diagram of Figure 20.



Figure 20 - Flow of Subtraction Algorithm

The LED over the lens of the camera is first turned on. While the manipulator and camera are steady, an image is then grabbed by the imaging system. The LED is then turned off. A second image is grabbed by the imaging system. The "off" image is then subtracted from the "on" image.

This results in an image with minimal noise.

To get the required information for the control, this image must be analyzed using the blob analysis software. This is referred to as "Get Blobs" in the software. The Get Blobs routine uses the subtraction algorithm to produce an image that the blob analysis software can more easily interpret. The blob analysis software processes the image and produces an array of structures. Each element of the structure represents a blob. The relevant attributes of the structure are the location of the centroid of each blob and its area. A sorting of the blobs, from largest to smallest, distinguishes between the LED (the largest) and the retroreflectors. This uses a "bubble" sort to rearrange the order of the array of structures so that the largest blob's structure is in the first element of the array of structures. Figure 21 diagrams this process.

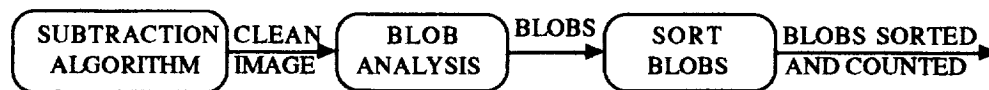


Figure 21 - Flow of Get Blobs

The initial target recognition algorithm is outlined as follows in Figure 22.

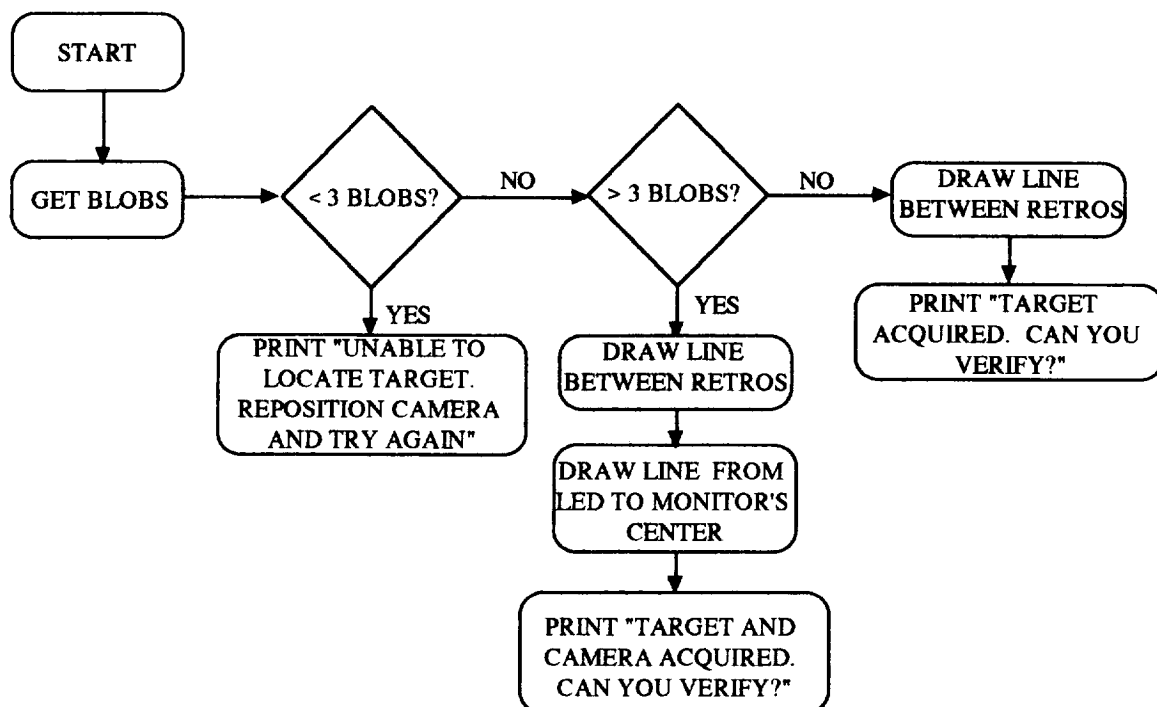


Figure 22 - Algorithm to Implement Initial Target Recognition

The initial target recognition implements the Get Blobs algorithm and then checks for the number of blobs. If there are less than three blobs then either the relative angle between the camera and the mirror is too large or the retro-reflectors are out of the camera's field of view. The program then asks the user to reposition the camera. If three blobs are found a line is drawn on the monitor between the blobs and the user is asked to verify that the target has been found before proceeding. This should be the perimeter of the retroreflectors. If more than three blobs are found, the system assumes that the LED is visible. A line is drawn from the largest blob to the monitor's center and a line is drawn between each of the retroreflectors. The user is then asked to verify that the target and the camera have been identified.

Initial target recognition requires that the user can see a live image and see a representation of the imaging system's perception of the target position as marked by the retroreflectors. This is implemented by using two video monitors - one with the imaging system's output and another with the live camera output (Figure 23).

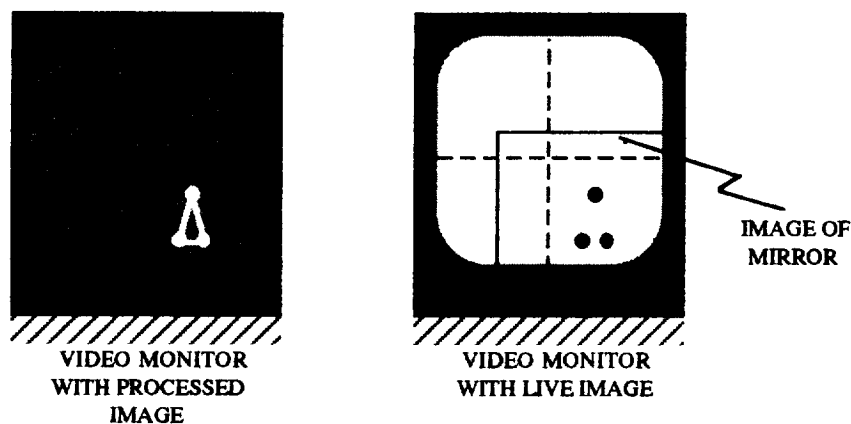


Figure 23 - Possible Scenario for Verification of Target Identification

Once the target has been identified, the user can be confident that the system will perform properly. The next step is to try to position the camera at a specific distance from the target to allow the FIND algorithm to attain an approximate normal more easily. This is referred to as "Initial Target Approach" and is outlined as follows.

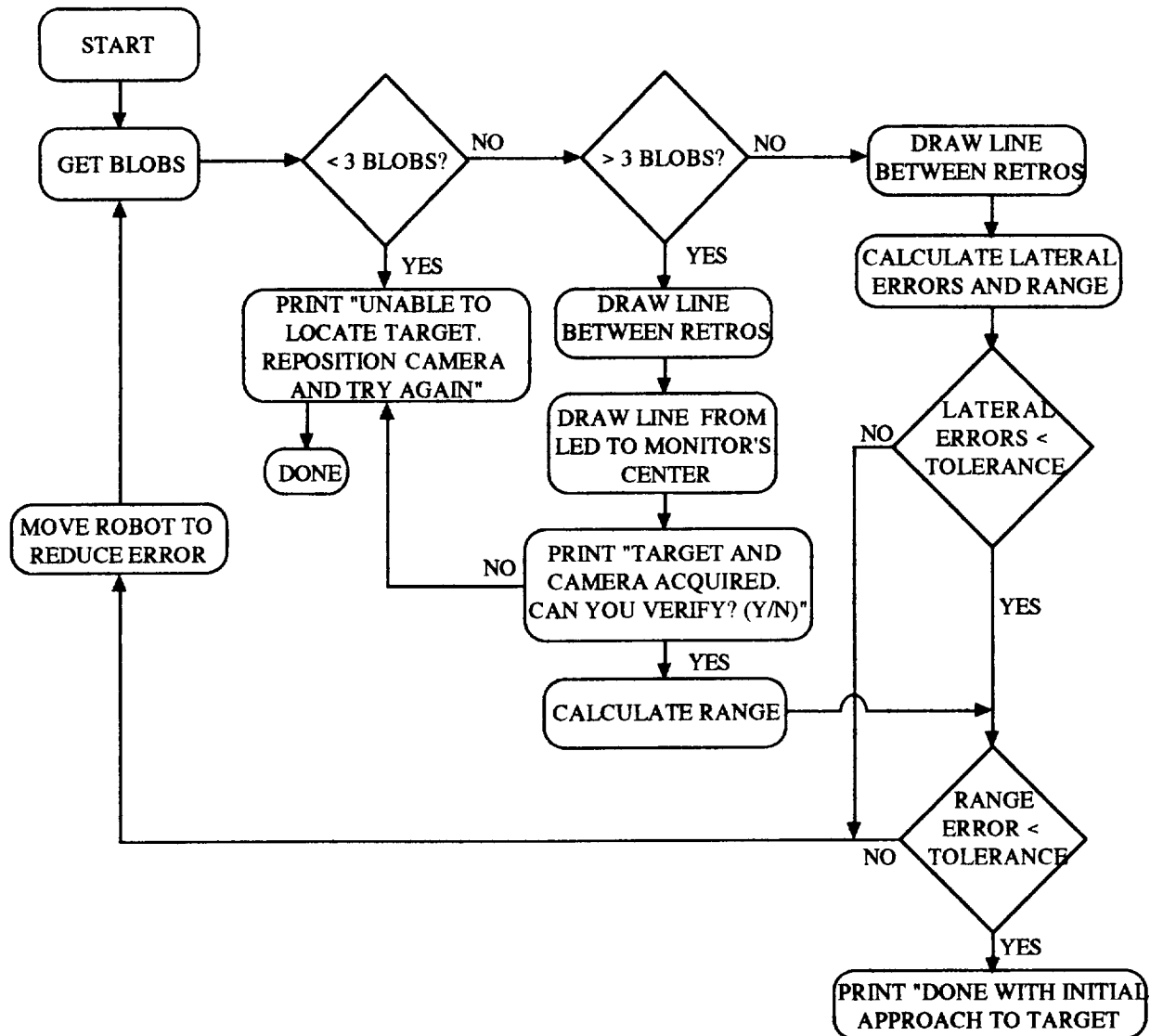


Figure 24 - Flow of Initial Target Approach

This algorithm first employs the Get Blobs algorithm. The next step depends on whether the Get Blobs routine provided less than three blobs, three blobs, or more than three blobs.

If less than three blobs are sensed, there has either been an error in the blob analysis or the retroreflectors are not visible to imaging system. In this case the user is told to reposition the camera and try again.

If more than three blobs are found, then the LED is visible in the reflection or some form of noise has entered the system. In this case, the system asks for help. If the user agrees that the LED is visible then the range error is minimized. This is accomplished by first determining the range error. If it is within the desired tolerance the initial approach to target is done. Otherwise the error is

used to determine the time delay for the range motion. If the user does not agree that the LED is visible then the system asks the user to reposition the camera and try again or another picture is needed without noise.

If three blobs are found, the retroreflectors are centered in the camera's field of view and then the camera is positioned at a specific range. This is accomplished by determining if the lateral error is greater than a set tolerance, usually 10 pixels. If the error is greater than the tolerance then the greatest error is reduced by a motion of the robot. This procedure is repeated until both lateral errors are within the tolerance. The range error is then concentrated on. It is nulled in the same manner as lateral errors.

The find algorithm is then implemented to put the camera into position to sense its own image in the mirror. Some sub-algorithms must be first explained that facilitate the description of the find algorithm. The first of these has already been used within other routines. It will be referred to as "Check for Retros and LED" and is shown in Figure 25.

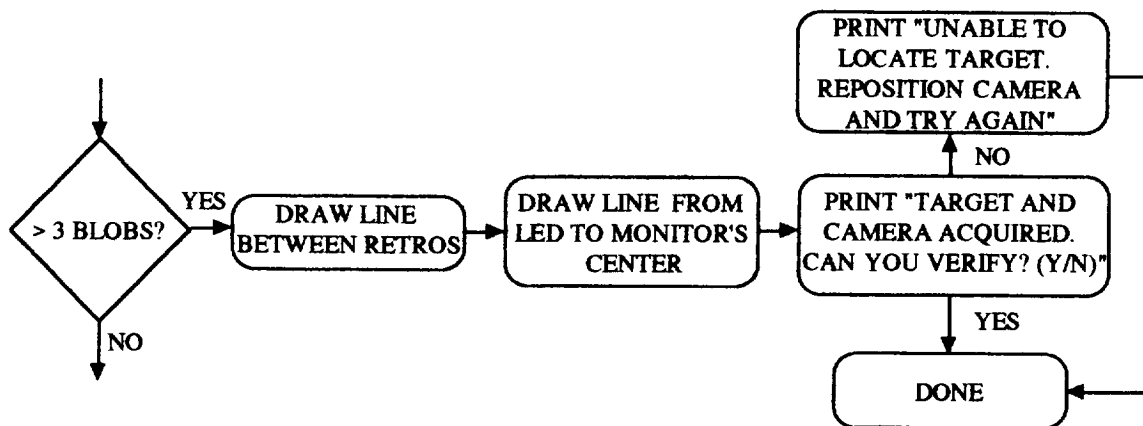


Figure 25 - Flow Chart of "Check for Retros and LED"

The execution of the algorithm will only be noticed if more than three blobs are found in the image by the imaging system. If this condition is true, lines are drawn connecting what the imaging system perceives to be the retroreflectors and another line connecting the center of the image and the LED. The user is then asked whether this is indeed the case. If the imaging system is correct it returns control back to the user with a message signifying the find algorithm is completed. If an

error has occurred, the user is prompted to re-align the camera and try the algorithm again.

A second necessary sub-algorithm will be referred to as "DO ROTATION". Its flow is shown in Figure 26.

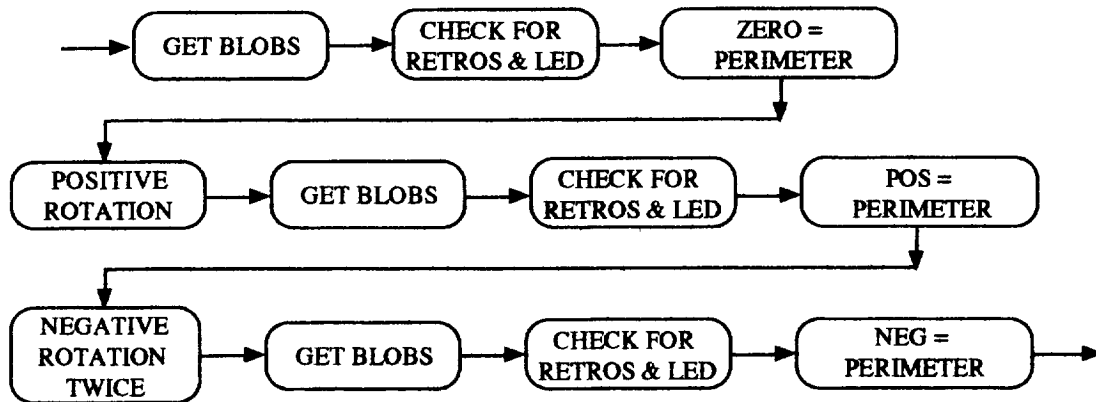


Figure 26 - Flow Chart of DO ROTATION

This routine calculates a reference perimeter size (ZERO), a perimeter size after a positive rotation (POS) and after a negative rotation (NEG). The variable, ROTATION, must be set to either PITCH or YAW before calling it. The perimeter is calculated using the blob's positions. Consider the blobs to be points in a plane, POINT 1, POINT 2, and POINT 3, having positions Y_1, Z_1, Y_2, Z_2, Y_3 , and Z_3 , as shown in Figure 27.

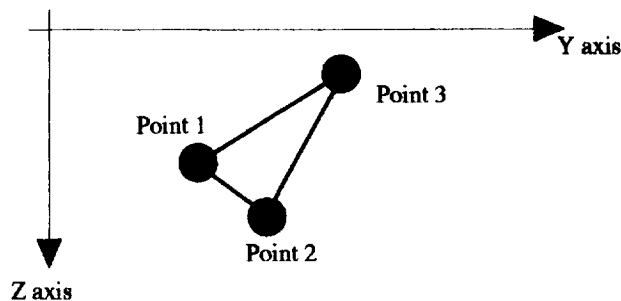


Figure 27 - Perimeter Calculation

The perimeter is then the sum of the lengths of the lines connecting the points as in equation 9

$$PER_p = \sum_{i=1,2,3; j=2,3,1} \sqrt{(Y_i - Y_j)^2 + (Z_i - Z_j)^2} \quad (9)$$

These sub-algorithms make it possible to implement the "FIND" algorithm as shown in Figure 28.

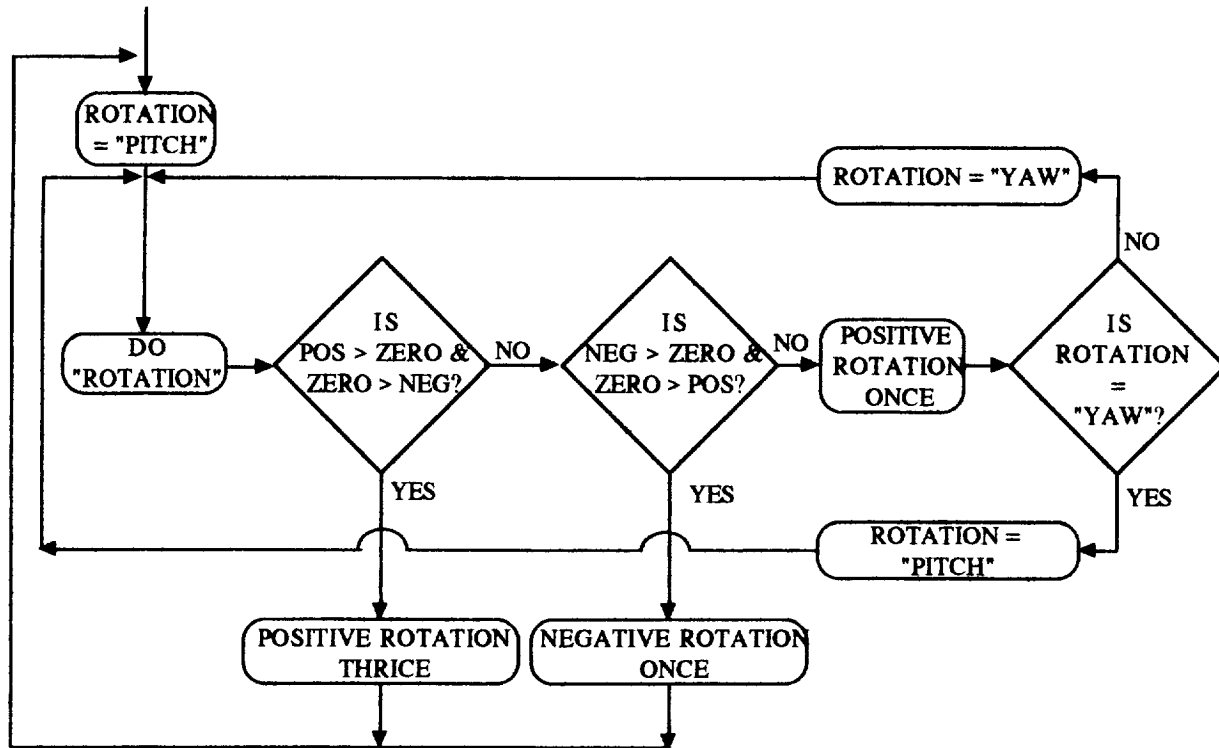


Figure 28 - Flow Chart of Find Algorithm

The algorithm is based on the optical property that the perimeter will be largest when the camera's optical axis is perpendicular to the target. The explanation of this can be facilitated by visualizing a triangle in three-dimensional space. If its plane is perpendicular to the line of sight it will have its maximum apparent size. As it is rotated away from perpendicular its apparent size decreases.

The algorithm is accomplished by performing a positive and a negative pitch motion. The apparent size of the retroreflector triangle perimeter is recorded before and after each motion. The robot is then commanded to move in the direction which causes an increasing perimeter size. This procedure is repeated for the yaw motion. The yaw and pitch motions are alternately implemented thus causing the alignment to be optimized. This is continued until a fourth blob is seen by the

camera. The user verifies that the fourth blob is the LED and then the fine positioning algorithm can take over.

The fine positioning algorithm allows for precision alignment of the end effector. A diagram of the flow of the program is shown in Fig. 29.

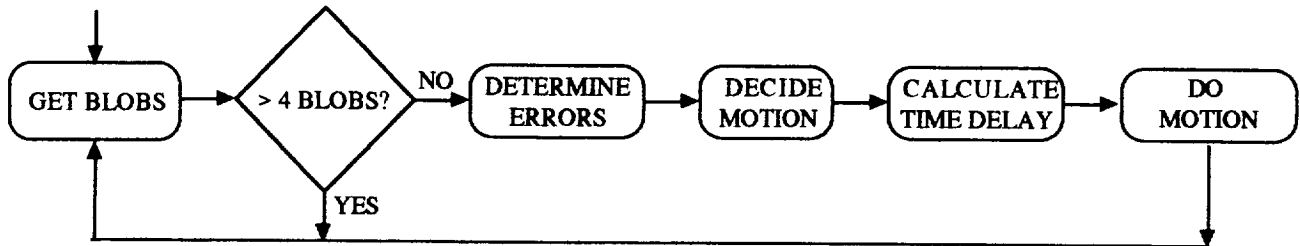


Figure 29 - Flow Chart of the Fine Positioning Algorithm

This routine first implements Get Blobs and checks to make sure that exactly four blobs were sensed. If not, the user is told that the target and camera cannot be acquired and asked if Get Blobs is to be repeated. Once a good image has been acquired the six errors are determined. A routine then uses these errors to determine which motion the robot should take. The pitch or yaw errors are minimized first, based on which is largest. The lateral errors are then minimized, followed by the range and finally the roll.

After the motion has been decided, the time delay for the motion is calculated and finally the motion is performed. The algorithm is designed to stop when the errors are all within a specific dead band. The dead band is dependent on the minimum possible motion of the robot in each direction, the resolution of the image and several other constraints that will be discussed in greater detail later.

The Complete System

Now that all the components of the system are developed, the system operation can be described. The automation is implemented by a user-driven menu where the user guides the progression of the automation. Each algorithm is available for user-implementation by pressing the key on the PC keyboard corresponding to the option of interest. All of the options listed are previously described except the manual control. This employs the interface between the PC and the teach pendant to allow the PC to simulate the teach pendant. The keyboard replaces the teach pendant so the user can manually control the robot by pressing various keys to activate the various

motion axes.

Performance of the Automation

The following section examines the expected performance of the automated TRAC system in terms of accuracy, resolution and repeatability.

Uncertainty Analysis

The dead bands of the controller show the performance of the system . An uncertainty analysis is necessary to obtain a real understanding of the accuracy of the numbers. Accuracy is defined as how close to exact alignment the system is capable of. It is therefore desirable to determine the end-effector's error with respect to the absolute position. The composite error is made up of the six directional errors. The errors of interest are the deviation of x , y , z , roll, pitch and yaw, as previously defined, from their positions as reported by the computer algorithm.

The uncertainty of a system, ω_s , is a measure of the probable maximum deviation from a reported value. It is defined as the square root of the sum of the products of the change in the dependent variable with respect to each independent variable and the uncertainty of the independent variables ω_i [7].

$$\omega_s = \left[\left(\frac{\partial D}{\partial I_1} \omega_{I_1} \right)^2 + \left(\frac{\partial D}{\partial I_2} \omega_{I_2} \right)^2 + \dots \right]^{1/2} \quad (10)$$

where I_i and D are the independent and dependent variables, respectively.

Before the six errors of interest are determined some preliminary uncertainties must be derived. The first is the uncertainty of the centroid of the retroreflectors in pixels. The equation for the centroid in each of the y and z directions is identical. For the y direction the centroid equation is as follows.

$$\text{CENT}_y = \frac{(y_1 + y_2 + y_3)}{3} \quad (11)$$

where y_1 , y_2 , and y_3 are the three retro-reflector positions in the y direction. In equation 11, the

y_i are the independent variables and $CENT_y$ is the dependent variable. With the partial derivatives of equation 11 in equation 10 and assuming the uncertainty of each centroid, ω_{y_i} , is 0.5 pixels, the centroid uncertainty ω_{CENT_y} becomes 0.289 pixels.

The uncertainty of the perimeter measurement in millimeters is next determined. The positions of the retro-reflectors on the mirror are described in Figure 30.

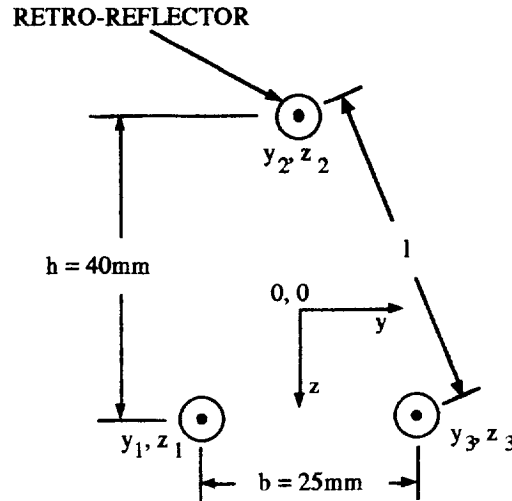


Figure 30 - Retro-reflector positions about the center of the mirror

The perimeter defined by the retro-reflector triangle can be shown to be

$$PER_m = 2 \left[h^2 + \left(\frac{b}{2} \right)^2 \right]^{1/2} + b \quad (12)$$

Thus $PER_m = 108.8$ mm. With the derivatives of equation 12 in equation 10 where h and b are the independent variables and PER_m is the dependent variable, and with $\omega_h = \omega_b = 0.5$ mm

$$\omega_{PER_m} = 1.33 \text{ mm} \quad (13)$$

This is the uncertainty in the perimeter in millimeters.

This calculation can then be built upon to produce the uncertainty of the perimeter measurement in pixels. The uncertainty of the perimeter in pixels from equation 10 is:

$$\omega_{PER_p} = \left[\left(\frac{\partial PER_p}{\partial z_1} \omega_{z_1} \right)^2 + \left(\frac{\partial PER_p}{\partial z_2} \omega_{z_2} \right)^2 + \left(\frac{\partial PER_p}{\partial z_3} \omega_{z_3} \right)^2 + \left(\frac{\partial PER_p}{\partial y_1} \omega_{y_1} \right)^2 + \left(\frac{\partial PER_p}{\partial y_2} \omega_{y_2} \right)^2 + \left(\frac{\partial PER_p}{\partial y_3} \omega_{y_3} \right)^2 \right]^{\frac{1}{2}} \quad (14)$$

$\omega_{y_i} = \omega_{z_i} = 0.5$ pixels and the partial derivatives are found from equation 9.

To enable this uncertainty to be calculated, numerical values are needed for the retroreflector positions because this particular uncertainty is a function of range. Since the aligned position is the position of interest for this application, it will be used for the calculation. (By aligned position, it is meant that the centroid of the retroreflector pattern is centered in the monitor image.) The location of the retro-reflectors is also dependent on the range. If a range of 350 millimeters is used then the perceived perimeter PER_p is 409.67 pixels and the aligned position is

$$\begin{aligned} y_1 &= -47.35 \text{ pixels}; \quad z_1 = 50.50 \text{ pixels}; \\ y_2 &= 0.00 \text{ pixels}; \quad z_2 = -101.01 \text{ pixels}; \\ y_3 &= 47.35 \text{ pixels}; \quad z_3 = 50.50 \text{ pixels}. \end{aligned} \quad (15)$$

With this data in equation 14, the perimeter uncertainty becomes

$$\omega_{PER_p} = 1.4864 \text{ pixels} \quad (16)$$

Range Uncertainty

The x or range calculation can be analyzed by using equation 4 in equation 10. The range uncertainty is then

$$\omega_{RANGE} = \left[\left(\frac{\partial RANGE}{\partial PER_p} \omega_{PER_p} \right)^2 + \left(\frac{\partial RANGE}{\partial A} \omega_A \right)^2 + \left(\frac{\partial RANGE}{\partial B} \omega_B \right)^2 \right]^{\frac{1}{2}} \quad (17)$$

where A and B were previously presented, $\omega_A = 578.$, and $\omega_B = 2.49$. Numerically, equation 17 becomes

$$\omega_{RANGE} = 3.030 \text{ mm}. \quad (18)$$

Lateral Position Uncertainty

The lateral position (y and z) uncertainty is derived using equation 1 in equation 10. The uncertainty is then

$$\omega_{ERR_m} = \left[\left(\frac{\partial ERR_m}{\partial ERR_p} \omega_{ERR_p} \right)^2 + \left(\frac{\partial ERR_m}{\partial PER_m} \omega_{PER_m} \right)^2 + \left(\frac{\partial ERR_m}{\partial PER_p} \omega_{PER_p} \right)^2 \right]^{\frac{1}{2}} \quad (19)$$

where

$$\frac{\partial ERR_m}{\partial ERR_p} = \frac{PER_m}{PER_p}, \quad \frac{\partial ERR_m}{\partial PER_m} = \frac{ERR_p}{PER_p}, \quad \frac{\partial ERR_m}{\partial PER_p} = -ERR_p \frac{PER_m}{PER_p^2}$$

With $\omega_{ERR_p} = 0.5$ pixels and $ERR_p = 0$ due to the aligned position,

$$\omega_{ERR_m} = 0.1328 \text{ mm} \quad (20)$$

Rotational Uncertainty (Pitch/Yaw)

Equation 6 is used to derive the uncertainty calculation for the pitch and yaw positions. The resultant uncertainty equation is

$$\omega_{\theta} = \left[\left(\frac{\partial \theta}{\partial ERR_m} \omega_{ERR_m} \right)^2 + \left(\frac{\partial \theta}{\partial RANGE} \omega_{RANGE} \right)^2 \right]^{\frac{1}{2}} \quad (21)$$

where

$$\frac{\partial \theta}{\partial ERR_m} = \frac{-1}{\sqrt{RANGE^2 - ERR_m^2}} \quad \text{and} \quad \frac{\partial \theta}{\partial RANGE} = \frac{-ERR_m}{RANGE \sqrt{RANGE^2 - ERR_m^2}}$$

With $ERR_m = 0$, $RANGE = 350$ mm, and equation 20, the uncertainty of the pitch or yaw position is

$$\omega_{\theta} = 3.79 \cdot 10^{-4} \text{ rads.} = 0.022 \text{ degs.} \quad (22)$$

Roll Uncertainty

Finally the uncertainty of the roll calculation is derived from equation 7. The uncertainty is then

$$\omega_{\phi} = \left[\left(\frac{\partial \phi}{\partial \text{ERR}_y} \omega_{\text{ERR}_y} \right)^2 + \left(\frac{\partial \phi}{\partial \text{ERR}_z} \omega_{\text{ERR}_z} \right)^2 \right]^{1/2} \quad (23)$$

where

$$\frac{\partial \phi}{\partial \text{ERR}_y} = \frac{\text{ERR}_z}{\text{ERR}_y^2 + \text{ERR}_z^2} \quad \text{and} \quad \frac{\partial \phi}{\partial \text{ERR}_z} = \frac{-\text{ERR}_y}{\text{ERR}_y^2 + \text{ERR}_z^2}$$

At alignment for a range of 350 mm, $\omega_{\text{ERR}_y} = .577$ mm, $\text{ERR}_y = 0$, and $\text{ERR}_z = 101$ mm the uncertainty is

$$\omega_{\phi} = 5.71 \cdot 10^{-3} \text{ rad.} = 0.33 \text{ deg.} \quad (24)$$

Implementation of the Automation

Several unforeseen problems have plagued the automation and for the most part they have been hardware problems. For example, the PUMA manipulator that was initially designated to be used in this development had a failure related to its controller. Therefore the development was relocated to another PUMA 560 manipulator. In addition, the imaging system does not perform as well as was hoped. While its features are very useful they are not very consistent. Even when the camera is motionless two consecutively acquired images are not always identical and this causes automation problems. The root of the problem seems to be the synchronization between the camera and the imaging system. The manufacturer cannot duplicate the conditions at their facility due to the complexity of the configuration. Therefore minimal technical help is available. The problem has not been solved but its effects have been lessened by means of an software error trapping routine that ignores bad images or restarts the image acquisition sequence any time errors are received from the image acquisition or the blob analysis routines.

Once the initial problems were overcome, the limits of the system were determined experimentally. The bounds of the controller's dead band for the six errors define the accuracy of the automation. The users of the system "squeezed" the limits of the dead bands of the controller for each direction. These limits have evolved over the course of the development of the automation and are due to the smallest increment that the PUMA can move or rotate under relay control. The six

dead band errors are shown in the following table.

Table 2 - Dead Bands for the Automation of TRAC

MOTION AXIS	DEAD BAND
LATERAL	0.25 mm
PITCH/YAW	0.05 deg
RANGE	0.10 mm
ROLL	0.10 deg

If the uncertainties for the aligned position and uncertainties using the dead bands of Table 2 are compared there is virtually no difference. The accuracy of the six positions are summarized in the following table.

Table 3 - The Accuracy of the Automation

MOTION AXIS	ACCURACY
LATERAL	0.25 ± 0.13 mm
PITCH/YAW	0.05 ± 0.022 deg
RANGE	0.10 ± 3.15 mm
ROLL	0.10 ± 0.49 deg

The values of Table 3 are dependent on the minimum motion of the robot in each direction, the resolution of the imaging system and the angle of view of the lens.

The optimization of the controller is limited by the minimum motion of the robot. If the minimum motion of the robot is infinitesimally small then the least possible steady state error is also infinitesimally small. Conversely, if the minimum motion of the robot is large then the steady state error will be large also.

The resolution of the imaging system plays an important role also. For example, in the range calculation of equation 4, the perimeter size is the independent variable. If the resolution of the imaging system is increased (more pixels to represent same image), then the perimeter as measured by the imaging system is more precise. This should produce a better estimate of the values of A and

B. Therefore the range uncertainty of equation 18 will be smaller due to the decrease in the values of ω_A and ω_B . The same effect is present in all six position calculations. The increasing accuracy of the pixel position translates into increasing positional accuracy. Also affecting accuracy is altering the angle of view of the lens which effectively alters the resolution of the image. If the resolution of the imaging system is held constant and the angle of view increased, then the resolution of the image is effectively decreased. The converse is also true.

In the matter of gross positioning, the critical step is the initial position of the camera with respect to the mirror at "target recognition." The reliability of the user to recognize that all three retro-reflectors are in the camera's view is essential. An additional consideration must be introduced. The nature of the FIND algorithm is such that there are certain positions where the user-verification may be successful but the routine has a poor chance of being successful. The main consideration is the capture angle of the retro-reflector. The retro-reflectors in use in this implementation (0.282 inch diameter, Technical Spec Corner Cube Reflectors from Edmund Scientific) have a capture angle of approximately sixty degrees or thirty degrees from the normal to the reflecting surface. Consequently, if the camera is aligned at approximately thirty degrees with respect to the mirror then the retro-reflectors will be visible. However, the FIND algorithm may have trouble performing its task. Figure 31 shows a possible scenario.

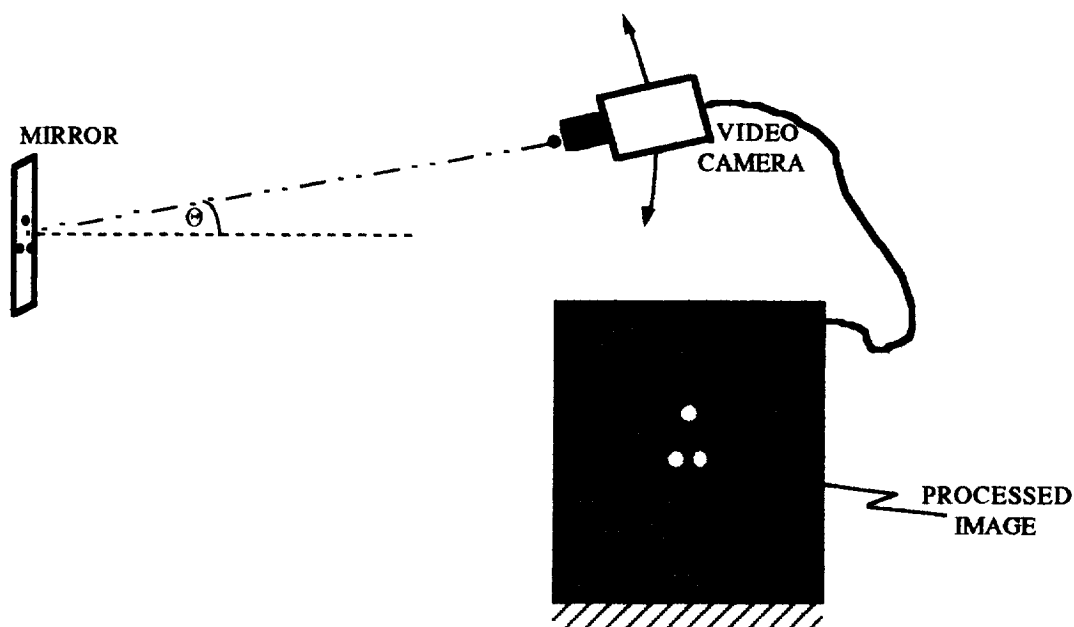


Figure 31 - Possible Problem with the FIND Algorithm

The image of the retro-reflectors can be seen at the initial position. If Θ is initially near thirty degrees, an ensuing motion causing the angle to increase will probably cause the retro-reflectors to lose their illumination and subsequently cause the routine to fail.

Considerations And Limitations

During the design and development of this automation several limitations of the system have become evident and must be mentioned. The description of each follows.

First of all, the subtraction routine is not 100% effective in eliminating exogenous noise. If the system experiences light bright enough to flood the lens then the subtraction algorithm is useless. This became evident in the testing of the automation. The lighting above the robot consists of banks of four fluorescent tubes. When the camera is in a position so that the tubes are directly behind the camera then these lights shine into the mirror and are reflected back to the camera. Subsequently, the lens is flooded with light. Therefore the system is inoperable if very bright lights can be encountered. It should also be noted that the definition of very bright is a function of each lens and how far the aperture can be closed. This possible problem can be mitigated however because hardware can be developed to shield the mirror from light emanating directly behind the camera.

A crude device was used to perform the shielding when background light interfered with the automation. The device was simply a small piece of cardboard approximately one foot square, mounted behind the lens but on the camera frame. This provided adequate shielding to any interference caused by objects behind the camera. If bright background lighting is a legitimate concern a more durable shield could be easily fashioned.

Another concern is the use of this system for long distance applications. This could be facilitated by more intense illumination of the target by the camera's light source. The current LED illuminates the retro-reflectors only if they are within approximately ten feet of the camera.

Problems arise when altering the light source. If the light source is placed laterally too far from the camera lens, the light from the retro-reflector can miss or be blocked from the camera lens by the iris or aperture and cannot be seen on the image plane of the camera as mentioned previously and illustrated in Figure 13. Therefore, the light source must be placed within a critical distance of the

lens to insure that the proper image of the retro-reflector is attained. An alternative to the LED is composed of a compound lens with a beam splitter and a beam spreader in the middle. Figure 32 shows what this might look like.

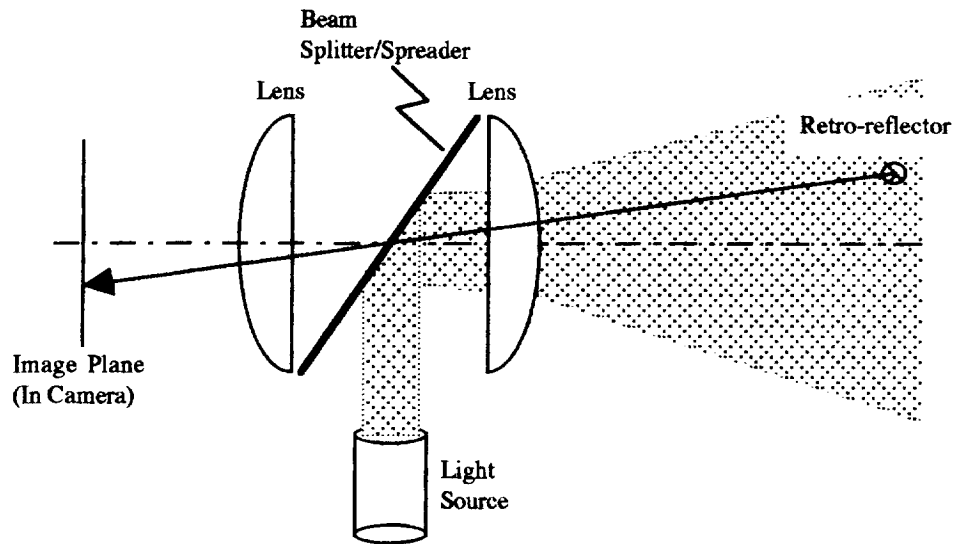


Figure 32 - Compound Lens with Beam Splitter/Spreader

The lens mechanism is essentially one lens split down the middle. Between the two lenses are a beam splitter and a beam spreader. It is also possible that the outer lens may work sufficiently as a beam spreader. This is analogous to putting a light source in the center of a single lens. This is a more complicated scenario and will be difficult to implement. The benefit of this system is that light reflected from the light source at any angle β within the capture angle of the retro-reflector will appear on the image plane (see Figure 13) .

Another alternative for long distance applications may be to use a bank of very bright LEDs around the perimeter of the lens and some less directional retro-reflective material on the mirror. Because of the larger distances involved and the relaxed directionality of the retro-reflectors, the light from the brighter LEDs should return as a cone and enter the lens of the camera.

Another limitation of the automation is related to the FIND algorithm and the angle of view of the lens. If the angle of view of the lens is increased then the PER_P calculation of equation 10 will be less accurate. The FIND algorithm is dependent on a consistent measure of this perimeter to perform as desired. For the distances and the lens tested this was not a problem but it is a potential problem upon other implementations.

As referred to previously, when the FIND algorithm is operating at the edge of the retro-reflector capture angle it has a better chance of failing. This could be alleviated by writing some smarter algorithms to take care of the case when one or several retro-reflectors suddenly disappear after a motion. This should signify to the controller that the previous motion was away from alignment so a motion in the opposite direction is necessary to attain alignment.

Bibliography

- [1] H.P. Bloch, "Use of laser optics for on-stream alignment verification," *Hydrocarbon Processing*, pp. 71-73, January, 1991.
- [2] H. P. Bloch, "Laser-optic Instruments Improve Machinery Alignment," *Oil and Gas Journal*, pp. 71-76, October 12, 1987.
- [3] H.P. Bloch, "Use Laser-optics for Machinery Alignment," *Hydrocarbon Processing*, pp. 33-36, October, 1987.
- [4] J. Callari, "Optical Alignment Keeps Sheet Running Straight," *Plastics World*, pp 27, 28, April, 1990.
- [5] G.T. Farnum, "Measuring with Lasers," *Manufacturing Engineering*, pp. 47-51, February, 1986.
- [6] M.R. Hamar and K.M. Gettelman, "Lasers Keep Machines In Line," *Modern Machine Shop*, pp. 92-102, February, 1991.
- [7] J. P. Holman, *Experimental Methods for Engineers*, Fourth Edition, McGraw-Hill Book Company, 1984, Chapter 3, pp. 46-59.
- [8] J. Ishikawa, "Accurate optical alignment with a single-mode optical fiber", *Applied Optics*, Vol. 25, No. 18, pp. 3013-3014, September 15, 1986.
- [9] M.F. Laguesse, "Optical potentiometer using fluorescent optical fiber for position measurement", *Applied Optics*, Vol. 28, No. 23, pp. 5144-5148, December 1, 1989.
- [10] H. S. Leibson, "Reduced Costs Help Ultrasonic Sensors Compete in Position-Detection Systems", *EDN*, pp. 5144-5148, November 28, 1985, pp. 75-82.
- [11] L. G. Monford, National Aeronautics and Space Administration, "Improved Docking Alignment System", *Official Gazette of the U.S. Patent & Trademark Office*, p. 211, January 2, 1990.
- [12] J. L. Moruzzi and R. Cunliffe, "Low-cost optical alignment system," *IEEE Proceedings*, Vol. 136, Pt. A, No. 2, pp 45-48, March, 1989.
- [13] T. Puntin, "Using Optical Techniques to Align Paper Machines," *Tappi Journal*", Vol. 70, pp. 179-81, May, 1987.
- [14] G.M. Robinson, "Smart Sensor Locates Target, Tracks its Movement," *Design News*, pp 86-88, September 23, 1985.
- [15] P. Saint-Marc, J.L. Jezouin, and G. Medioni, "A Versatile PC-Based Range Finding System", *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 2, pp. 250-256, April, 1991.
- [16] J. Shi, S. Monchaud, R. Prat, and J. Domey, "3D Object Localization and Automatic Tracking," *Advanced Sensor Technology*, pp. 197-207, 1988.
- [17] W.H. Stevenson, "Optical Sensors for Alignment, Distance Measurement, and Feature Location in Automated Manufacturing Operations," *Optical Sensing and Measurement*, pp. 87-96, 1988.